# SRIO Type 11 between two boards

## Purpose

The purpose of this lab is to demonstrate SRIO communication between two boards. A detailed description of the Lab is given in a separate power point presentation titled

***KeyStone SoC Training SRIO Demo: Board-to-Board***

## 

## Two EFVMs that are connected via a Break-Out-Board are needed to perform this Lab as is showed in the following picture;

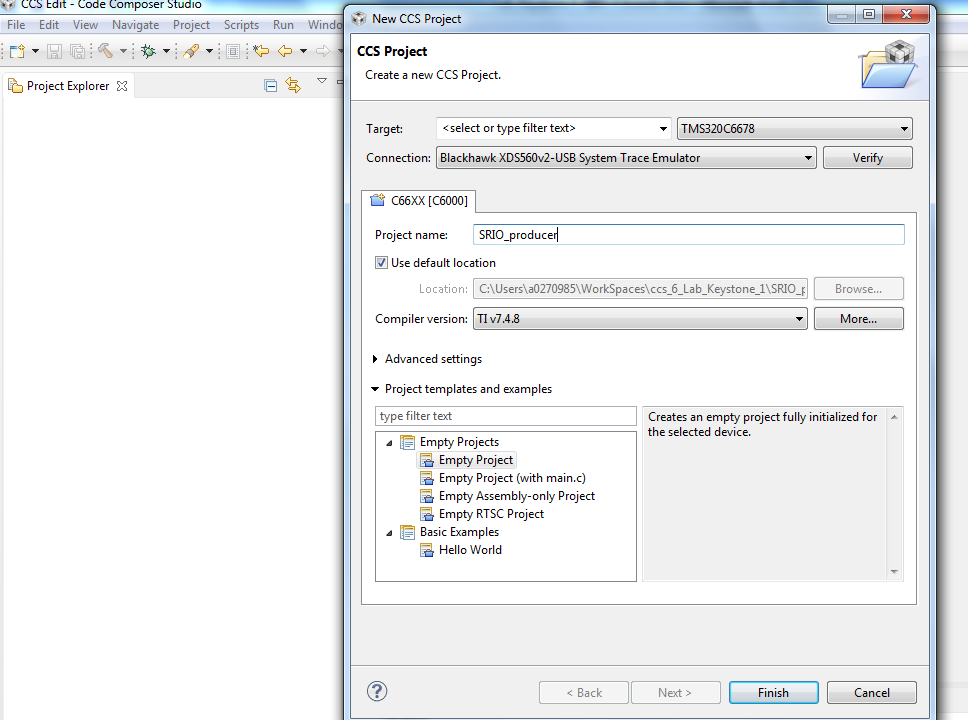
## IMG_4755

IMPORTANT NOTE:

The screen shots for this Lab are taken from CCS v6. CCS V5 might have slightly different interface.

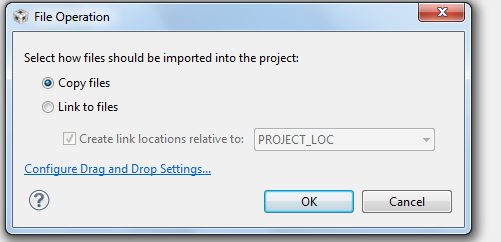
### Task 1: Producer - Start CCS new project

1. Start CCS. From the file tab select new->CCS Project and chose empty project with the name SRIO\_producer:

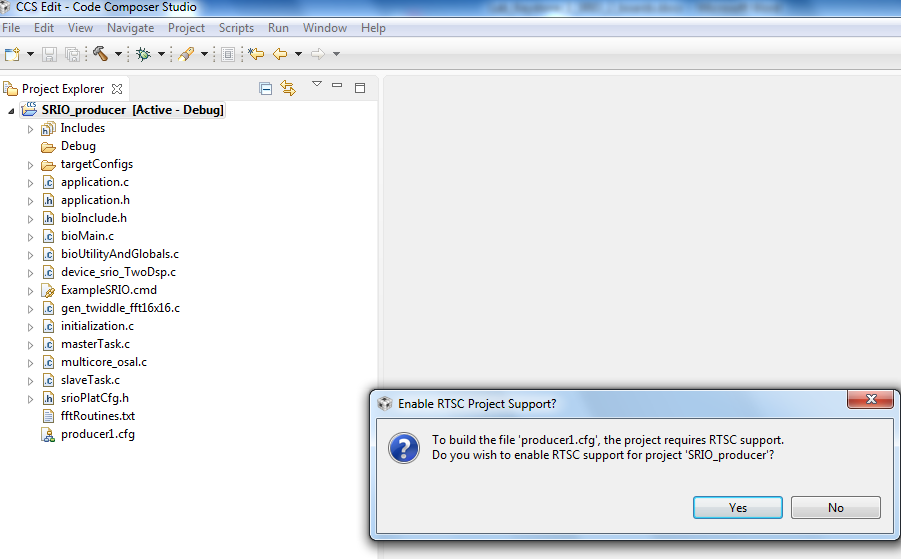


Click finish.

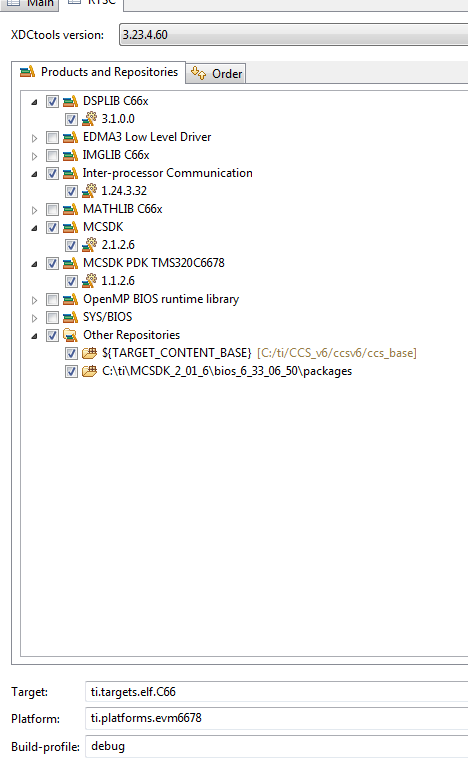
1. Select the new project, right click and choose add files. In the dialogue box navigate to the producer Original project location (the instructor will give the project location) and select all the files Click open. A dialogue box will ask you to copy or to link the files. Select copy and click OK.



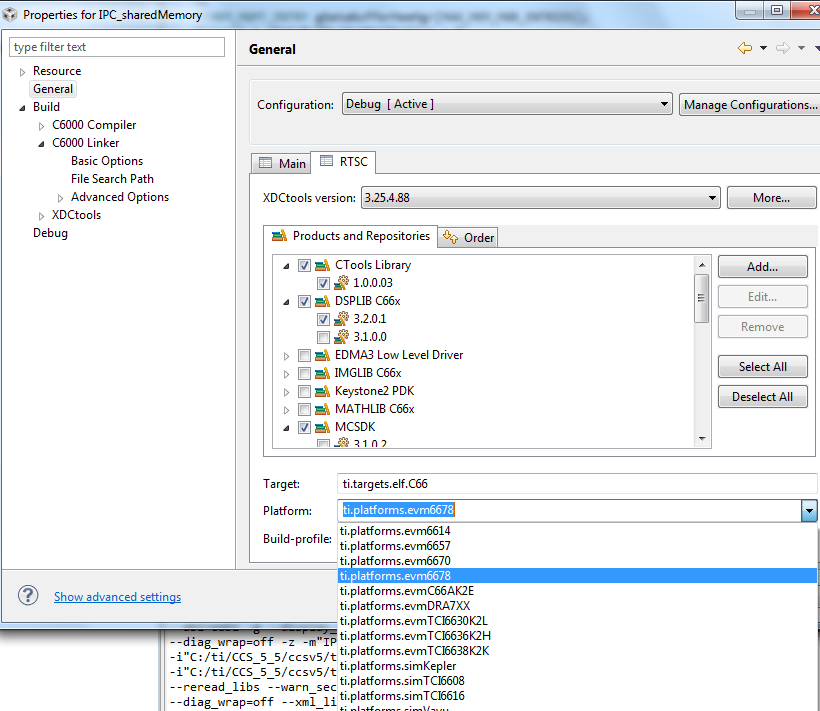
1. As soon as the files were copied into the project, a dialogue box will ask you if you want to start RTSC project. Choose yes and continue



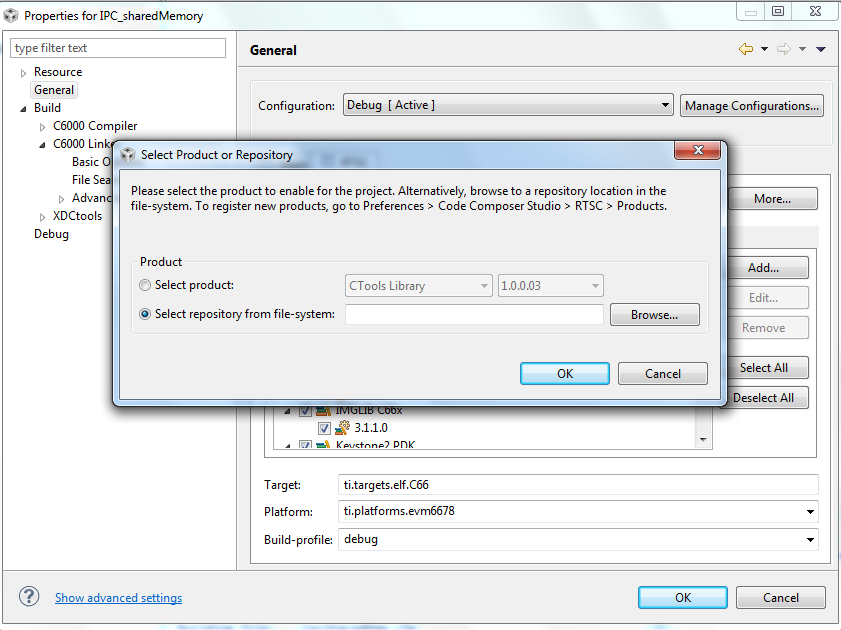
1. Next select the project, right click and select properties.
   1. In the RTC tab (general->RTSC) make sure that ONLY the repositories that are checked in the next screen shot are checked in your project



* 1. Make sure that the Target is ti.targets.elf.c66. If not, set it to ti.targets.elf.c66
  2. If the platform is not set, set the platform by starting the pull down menu (right click on the arrow next to the platform tab. Then choose ti.platforms.evm6678 by clicking twice on the correct platform

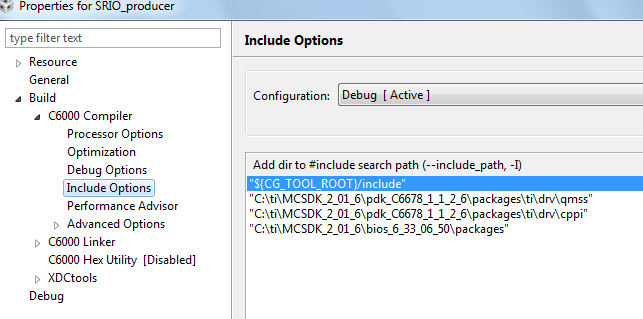


* 1. In the RTSC tab, add the ipc repository. From the RTSC window select click on ADD, and choose select repository from file system

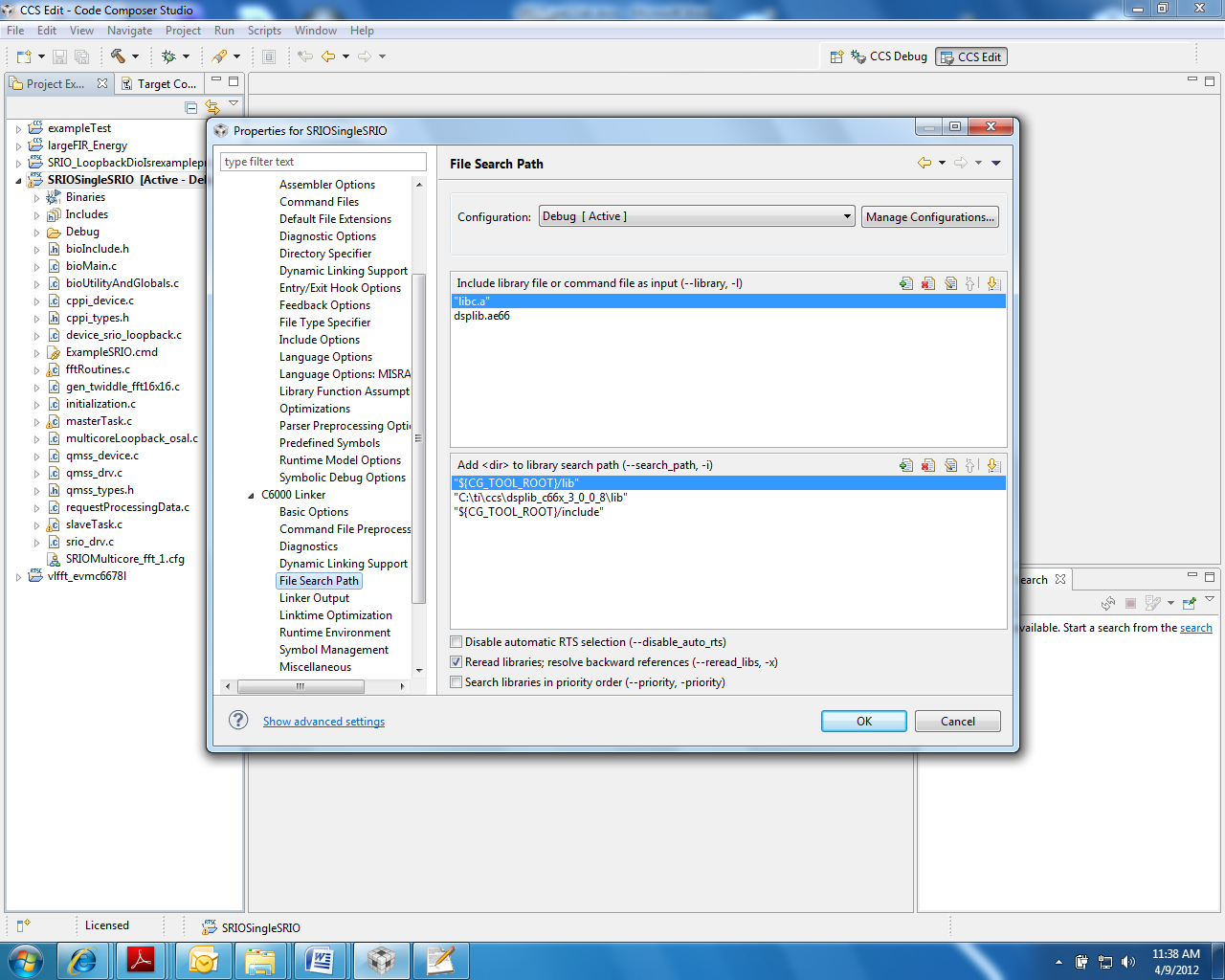


Browse to the location of the IPC packages. On my system it is in MCSDK\_2\_01\_6\ipc\_1\_24\_03\_32\packages. If your Laptop has a different release of MCSDK, the path will be changed accordingly

* 1. Click OK
  2. Click OK on the properties
  3. In the Include option add path to the cppi, qmss and bios. See the screen shot below. Your path to the pdk as well as the pdk version might be different. Search for the correct path

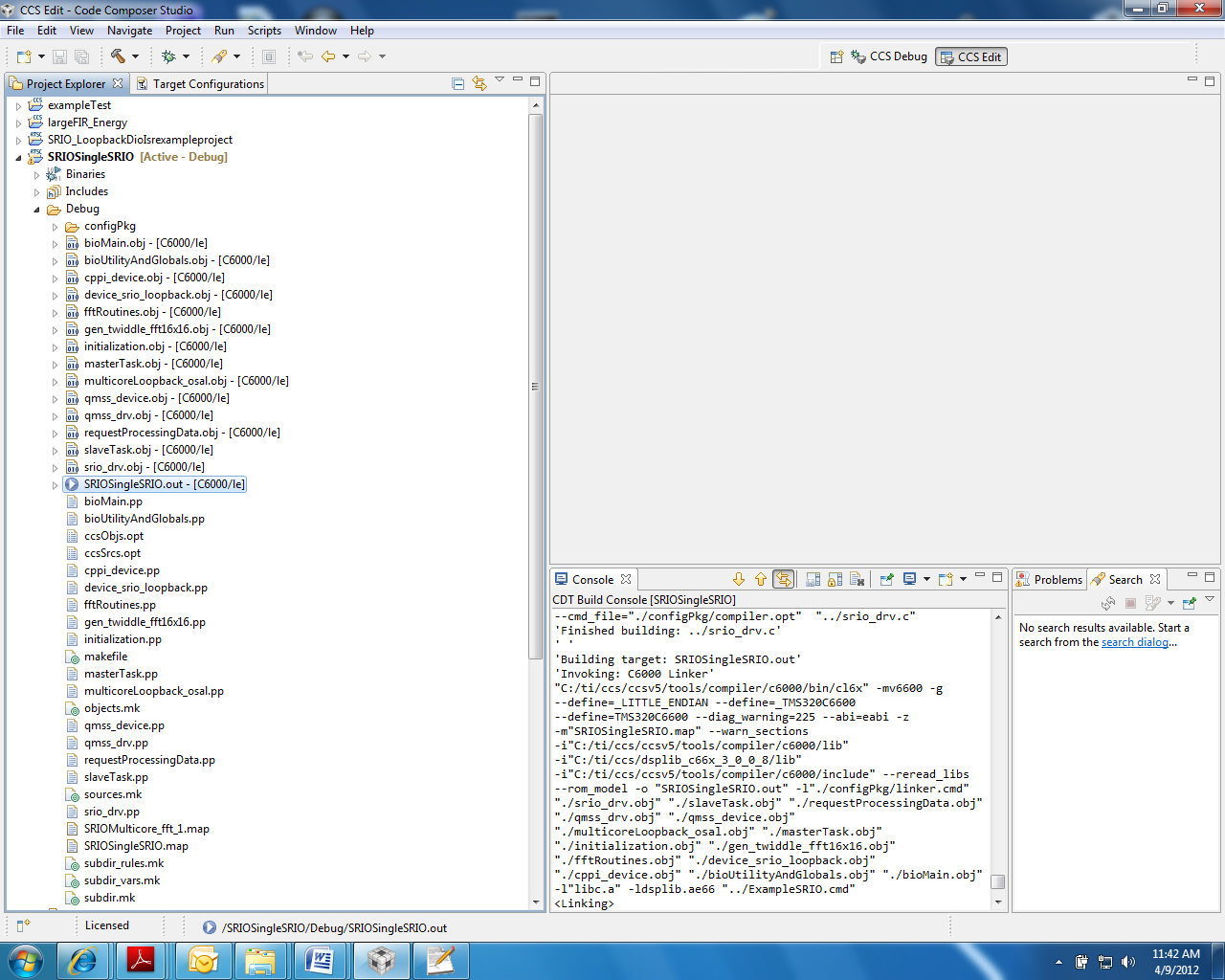


1. Do the same for the Linker File Search Path, as shown below. Add the library dsplin.ae66 in the upper window, add the path to the library in the lower window



### Task 2: Build the producer Application

1. Clean the build and build the project by selecting the project, click the right button and select re-build. The first build will take several minutes
2. Verify that the executable (.out) was built by looking at the debug directory (assuming the build configuration is debug configuration).



### Task 3: Consumer - Start CCS new project

1. Start CCS. From the file tab select new->CCS Project and chose empty project with the name SRIO\_consumer. (You can look at the screen shots from the producer instructions)

Click finish.

1. Select the new project, right click and choose add files. In the dialogue box navigate to the producer Original project location (the instructor will give the project location) and select all the files Click open. A dialogue box will ask you to copy or to link the files. Select copy and click OK.
2. As soon as the files were copied into the project, a dialogue box will ask you if you want to start RTSC project. Choose yes and continue
3. Next select the project, right click and select properties.
   1. In the RTC tab (general->RTSC) make sure that ONLY the repositories that are checked in the producer screen shot from above are checked in your project
   2. Make sure that the Target is ti.targets.elf.c66. If not, set it to ti.targets.elf.c66
   3. If the platform is not set, set the platform by starting the pull down menu (right click on the arrow next to the platform tab. Then choose ti.platforms.evm6678 by clicking twice on the correct platform
   4. In the RTSC tab, add the ipc repository. From the RTSC window select click on ADD, and choose select repository from file system

Browse to the location of the IPC packages. On my system it is in MCSDK\_2\_01\_6\ipc\_1\_24\_03\_32\packages. If your Laptop has a different release of MCSDK, the path will be changed accordingly

* 1. Click OK
  2. Click OK on the properties
  3. In the Include option add path to the cppi, qmss and bios

1. Do the same for the Linker File Search Path. Add the library dsplin.ae66 in the upper window; add the path to the library in the lower window.

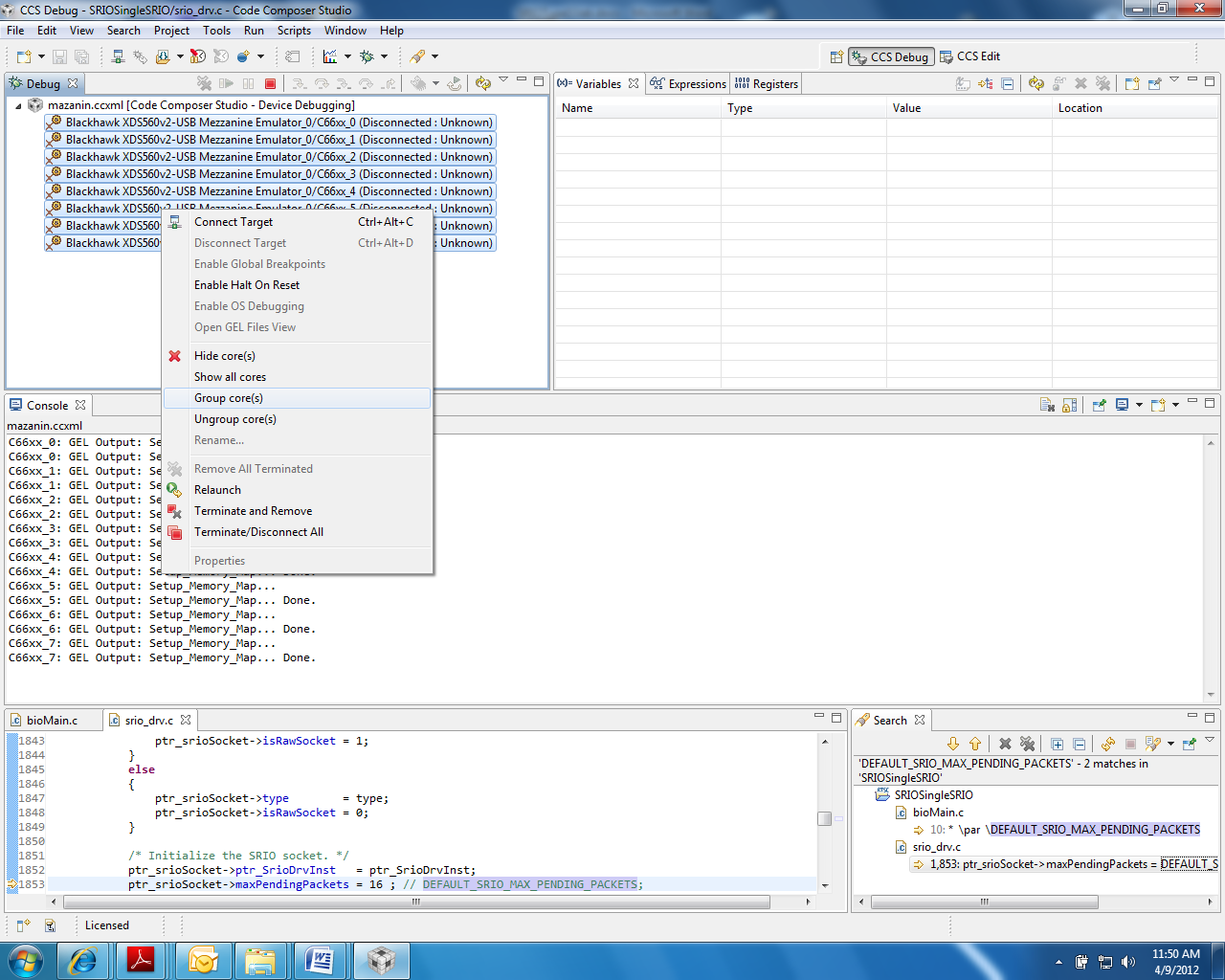
### Task 4: Build the consumer Application

1. Clean the build and build the project by selecting the project, click the right button and select re-build. The first build will take several minutes
2. Verify that the executable (.out) was built by looking at the debug directory (assuming the build configuration is debug configuration).

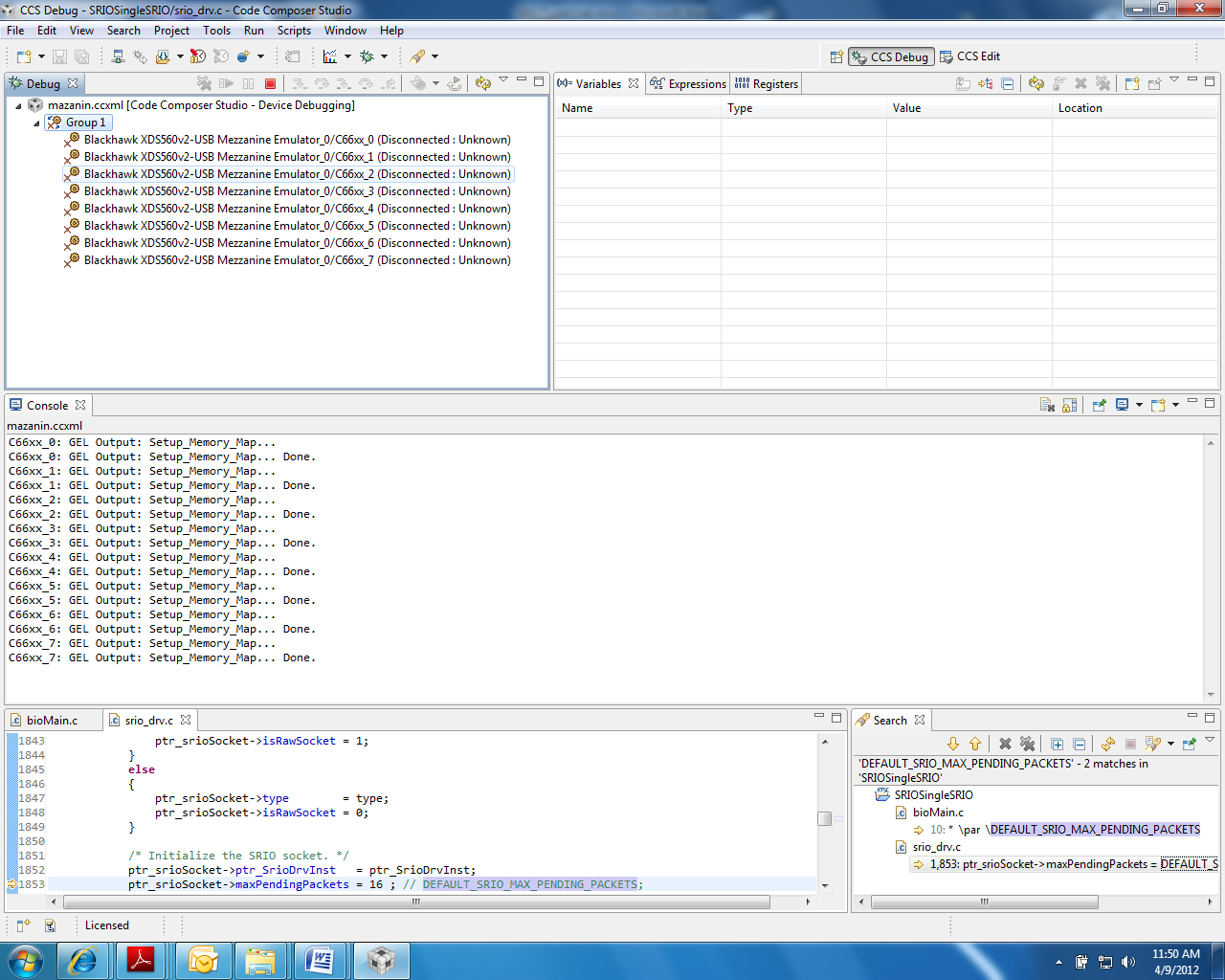
### Task 5: Launch the Debugger

The following instructions are for two laptops, each one is connected to an EVM board. For one laptop we have to define a new target that controls multiple boards. In this Lab we use two laptops.

1. Connect the mini USB side of the USB cable to the Mezzanine card on the top of the EVM. The mezzanine card is Blackhawk 560 emulator
2. Power on the two EVMs, connect each of the USB cable to the respected Laptop, wait for the EVM to finish boot (the red light is ON).
3. Launch your debugger. Instructions how to define target for 6678 are in the Appendix
4. CCS will change into the debug prospective
5. Group all cores into one group as follows:



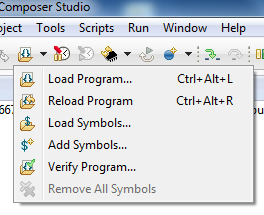
After grouping, Group 1 is defined and displayed as shown below:



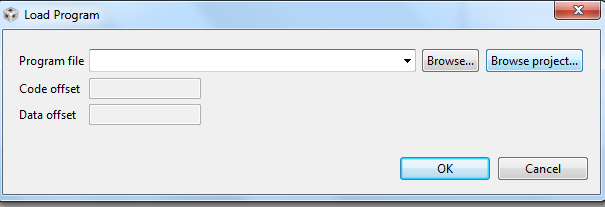
### Task 6: Load and Run two Laptops

The following instructions are for two laptops

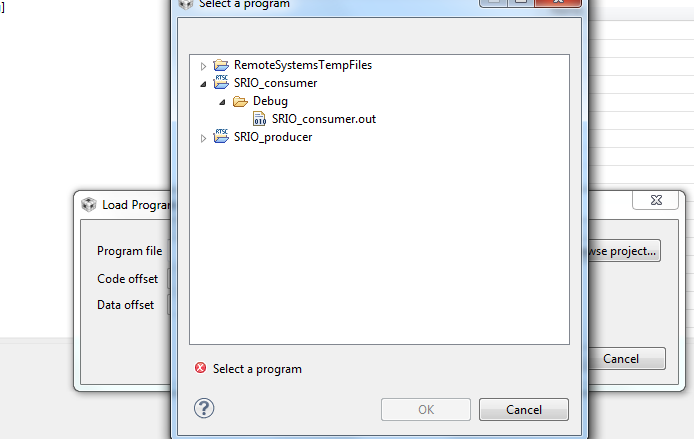
1. Select Group1, and connect all cores in the group by one of the three ways:
   * From the RUN menu, select Connect Target.
   * Right click on the group name and choose Connect Target.
   * Click the Connect Target icon.
2. One of the Laptop will load the producer code, one will load the consumer code
3. Load the code to all cores in the group:
   * From the RUN menu, select Load.  
     OR
   * Click the Load icon.



1. The easiest way to load the code is to choose Browse Projects and choose the right project:

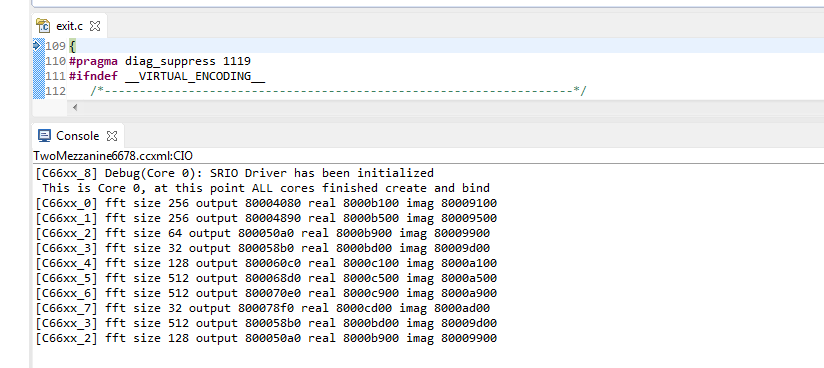


And then navigate to Debug tab, select the out file and click OK.



1. Run the code on both Laptops in one of the following ways:
   * Press F8.
   * From the RUN menu, select Resume.
   * Click on the Resume icon (green arrow).

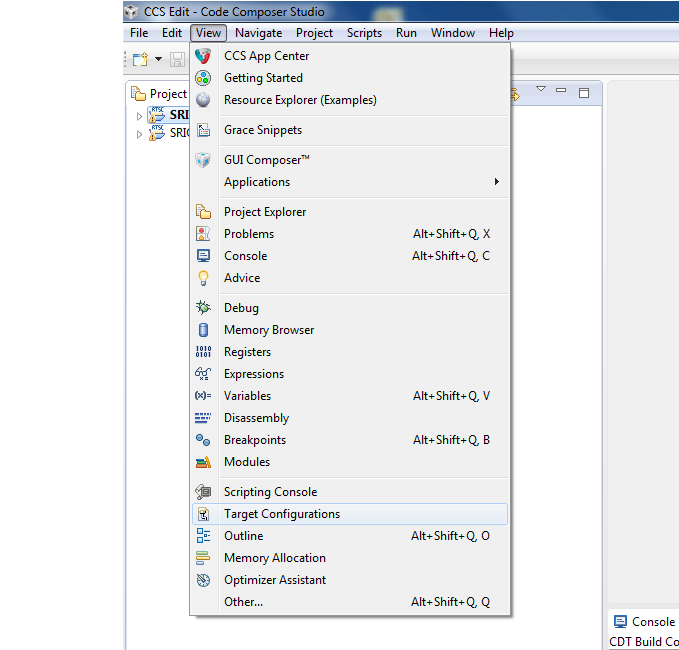
The output results appear on the consumer lap top similar to the follows:



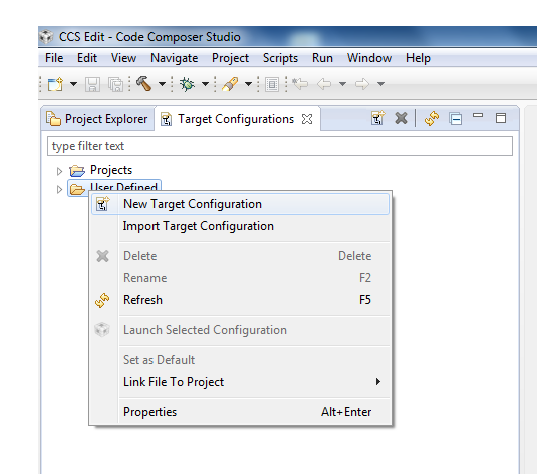
1. Observe the results, then suspend the run:
   * From the RUN menu, choose Suspend  
     OR
   * Click on the Suspend icon (the yellow “pause” lines)

# Appendix – Define target configuration

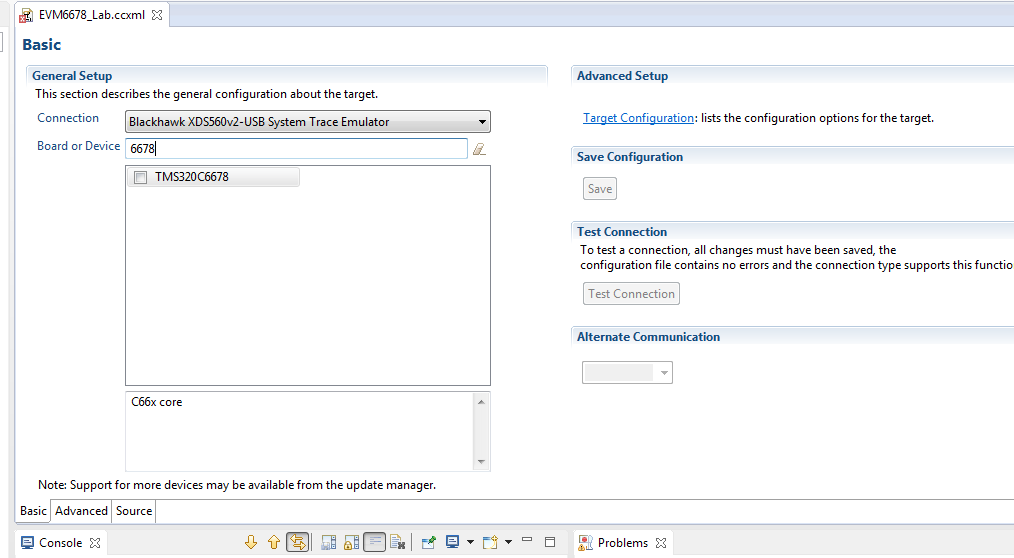
1. From the CCS Edit prospective click on the View tab, and from the pull down menu choose target configuration



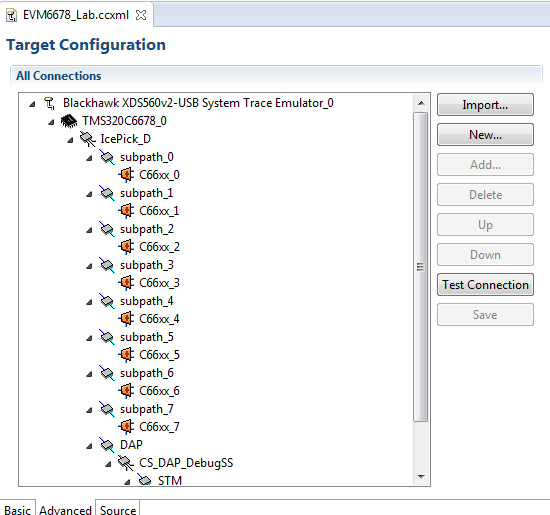
1. Right click on User defined tab and choose new target configuration



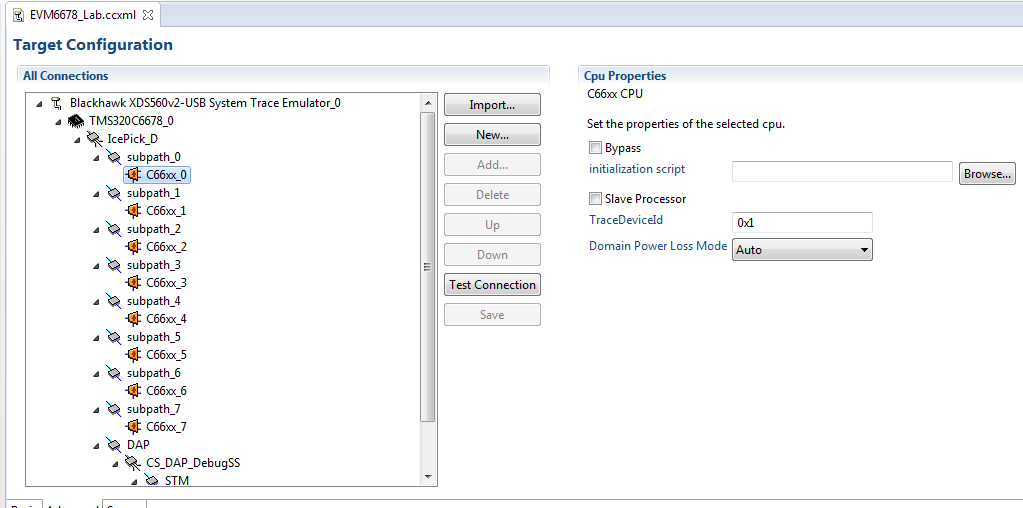
1. In the dialogue box that is opened give the name EVM6678\_Lab.ccxml and click finish.
2. Base window will open, choose the emulator type Blackhawk 560-v2 USB Trace emulator, and set 6678 as the filter as shown in the following screen shot:



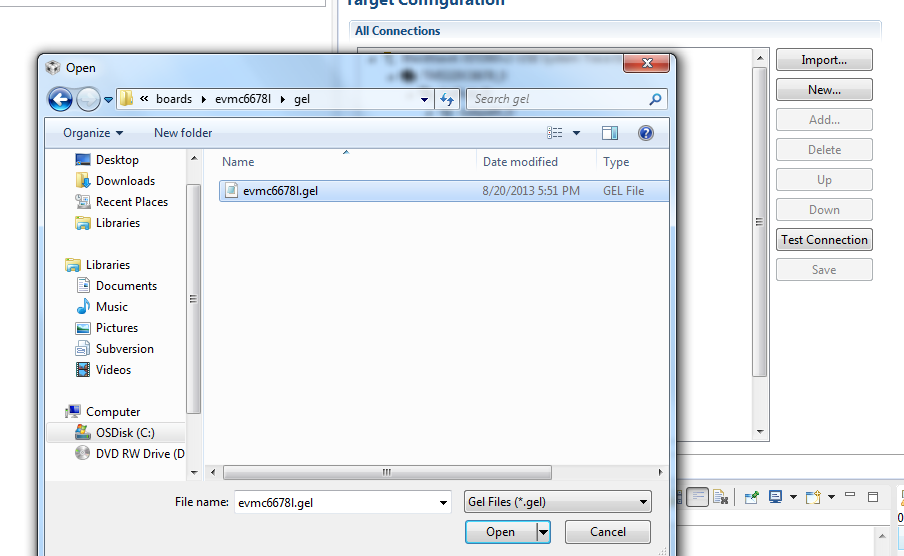
1. Check the TMS320C6678 and click save
2. Next we connect the gel file to the 8 cores. Click on advanced.



1. Click on the first core, C66xx\_0



1. Browse to the location of the gel file. Locate the CCS install directory (in my case, it is C:\ti\CCS\_v6, continue to \ccsv6\ccs\_base\emulation\boards\evm6678l\gel\evm6678l.gel and click open



1. Do the same to all other cores, select C66xx\_N where N goes from 1 to 8 and repeat the procedure from above. Note, after the first core, when you click on Browse the open window will be in the previously selected gel directory
2. After configuring all 8 cores with the gel file hit save. The target is ready. You can close the target window.